Swarm Based Neural Learning Algorithm for Data Classification

Devinder Kaur
University of Toledo
2801, W. Bancroft, Toledo, OH 43606
(419) 530-8163
devinder.kaur@utoledo.edu

Praneeth Nelapati
University of Toledo
2801, W. Bancroft, Toledo, OH 43606
(419) 530-8163
praneeth.nelapati@utoledo.edu

ABSTRACT
In this paper, the architecture of the neural network has been modeled using swarm particle representation. A new algorithm based on ACO and PSO is devised to adjust the parameters that include weights and thresholds. It has been shown that learning time of the neural network has been reduced in comparison with Neuro-Evolutionary and simple back propagation techniques. The classification error using the Neuro-Swarm approach is almost zero.

Categories and Subject Descriptors
I.2.m [Artificial Intelligence]–Miscellaneous - *Evolutionary computing and swarm intelligence.*

General Terms
Design, Algorithms, Performance

Keywords
ACO, PSO, swarm intelligence, neural networks, algorithm

1. INTRODUCTION
Swarm Intelligence is the technique of using multiple agents that communicate to solve a problem. In Swarming, each agent or particle is guided by its neighbors to the optimum solution. Two swarming techniques *Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO)* are considered efficient in optimization of problem domains. A new algorithm that incorporates features from ACO and PSO is designed to optimize the data classification domain by reducing the learning time in comparison to back propagation and Neuro-Evolutionary algorithms.

2. SWARM INTELLIGENCE
Swarm Intelligence is the discipline that deals with natural and artificial systems composed of many individuals that coordinate using decentralized control and self-organization. The discipline focuses on the collective behaviors that result from the local interactions of the individuals with each other and with their environment. The basic principle of swarming is very simple: by having a relatively large number of agents following very simple rules, complicated group-behaviors can be generated. The two distinct characteristics in the swarm modeling is that the agents have to work individually on the problem, while behaving as a group or single entity to optimize. Two significant areas in swarm research are *Ant Colony Optimization (ACO) and Particle Swarm Optimization* which is considered as efficient techniques for optimizing problem domains.

2.1. Ant Colony Optimization
Ants have highly structured social organizations. Several studies have been conducted to use this organization to solve complex real world problems [2], [4]. This kind of organization helps ants to achieve complex tasks like foraging and building nests. For instance, the foraging of ants is highly studied approach. The ants communicate by a chemical pheromone. When an ant starts the search for food, it leaves the pheromone in the path it travels. Thus, each ant may traverse different route to reach the food source. When a new ant starts the food search from the colony, it looks for the direction that has maximum pheromone trail, indicating that a considerable number of ants used this track for food [3]. As the pheromone evaporation takes place with time, the longer route has more probability of having weak pheromone trail while the shortest route has maximum probability of having strong pheromone trail. This situation suggests that the ants adjust their path finally to the shortest path.

2.2 Particle Swarm Optimization
PSO is another technique to solve real world problems that lack any mathematical model [5]. The technique involves in launching a group of agents randomly in the search space and assign some kind of measuring technique that assess whether a particle (Agent) is directed towards the global solution or not [5].

3. SWARM MODELING
The present section deals with the modeling of swarming techniques discussed in section 2 to suit the data classification domain.

3.1 Data Classification
To test the feasibility of the Neuro-Swarm algorithm, data classification domain with iris data as a bench mark has been chosen.

3.1.1. Iris Data Classification
Th***e*** Iris data set is a multivariate data set introduced by Sir Ronald Aylmer Fisher to describe discriminant analysis.

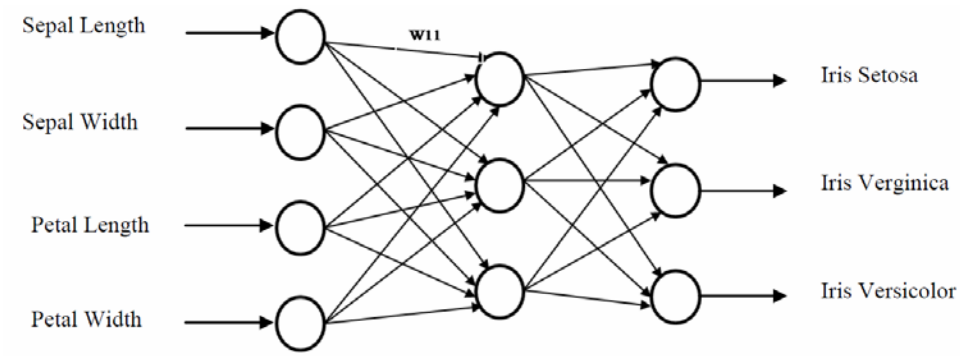


Fig 1: The Neural Network for the Iris Data problem

The iris data set is the collection of 50 samples each of three plant species namely Iris Setosa, Iris Versicolor and Iris Verginica. The data set consists of four features namely, sepal length, sepal width, petal length and petal

The problem is to identify the category of any iris flower based on its four input characteristics of sepal length, sepal width, petal length, petal width. Table 1 show an extract from the Iris data set, where a plant with a sepal length of 6.0 can be either a versicolor or a verginica. This indicates that each of the species has no distinguishable length and width ranges, based upon which the classification is done implying that any plant can only be classified by considering all its features which are inter-related making the classification more complex.

Table 1: Extract from the Iris Data Set

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.1	3.5	1.4	0.2	Setosa
6.0	3.4	4.5	1.6	Versi-color
6.0	3.0	4.8	1.8	Virginica

3.1.2. Neural Network

Neural Networks have been successfully used for iris data classification. Various algorithms are developed for neural network learning including back propagation [1], [8] and genetic algorithms [6]. In this paper, we are proposing a new swarm based neural network learning algorithms. Fig 1 shows the Neural Network for the Iris data classification. It has three layers; one input layer, a hidden layer and an output layer. The input layer has four neurons that correspond to the four plant characteristics of sepal length, width, and petal length, width. The hidden layer has three neurons. The output layer has three neurons corresponding to the three plant species, namely Iris Setosa, Iris Versicolor and Iris Verginica. A Neural Network stores the knowledge in the form of weights. The weights are to be properly adjusted so that the Iris plants are classified properly. The activation function used for the neural network is the sigmoid function. The sigmoid function can functionally be represented as

$$\mathbf{Sigmoid(x) = 1/\ (1+e^{-x})}$$

3.2 Characterizing the swarm particles

The next step is to define the structure and characteristics of the swarm particles. Fig 2 shows the structure of the swarm particle. The swarm particle is the ensemble of 27 individual elements; 21 elements representing the weights of the neural network and the rest of the 6 elements refer to the thresholds of the neurons in the neural network. The swarm algorithm creates a fixed number of swarm particles at random positions. Random positions refer to the random weights and thresholds that are assigned to each swarm particle. The aim of the algorithm is to reduce the mean classification error of the group of swarm particles to an acceptable value. The reason for choosing mean classification error (average of the classification error of the swarm group) over the best classification error (lowest classification error achieved in the swarm group) is to ensure that the solution obtained at the end is optimum. The choice of best classification error as the measure of stopping criterion may be misleading because the solution that the best particle has converged to may be a local maxima, indicating that the true solution is located elsewhere in the search space. The mean classification error ensures that the entire search space is properly explored as all the particles stationed at different points in the search space are contributing to the stopping criterion. However, the swarm particles have to work as individual units that collectively form a complete unit. To function as a collective unit, the swarm particles have to communicate and should follow few principles that govern the system. The search space is the collection of all the possible values that can be taken by weights and thresholds. The range of swarm particle's weights and thresholds must be properly defined, since this capability defines the extent of area the particle can explore. The search space for the algorithm has been defined as [0 1]. The choice of such a confined search space allows broadcasting; a good option for communication between the swarms. The Swarm particles follow the following principles in order to make themselves organized:

Homogeneity: All the swarm particles are created alike with a string of 27 real numbers (each representing a weight or threshold), because the goal of the algorithm is to reduce the classification error.

w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	w14	w15	w16	w17	w18	w19	w20	w21	w22	w23	w24	w25	w26	w27

w1:W11, w2:W12, w3:W13, w4:W14- Weights for the hidden neuron 1

w5:W21, w6:W22, w7:W23, w8:W24- Weights for the hidden neuron 2

w9:W31, w10:W32, w11:W33, w12:W34- Weights for the hidden neuron 3

w13:W51, w14:w52, w15:W53 - Weights for the Output Neuron 1

w16:W61, w17:w62, w18:W63 - Weights for the Output Neuron 2

w19:W71, w20:w72, w21:W73 - Weights for the Output Neuron 3

w22:T1 -Threshold of hidden Neuron 1

w23:T2- Threshold of hidden Neuron 2

w24:T3- Threshold of hidden Neuron 3

w25:T4-Threshold of output Neuron 1

w26:T5-Threshold of output Neuron 2

w27:T6-Threshold of output Neuron 3

Fig. 3: Chromosome for the Iris Data Classification

Locality: Locality defines the range within which the swarm particle can communicate other swarm particles. If the swarm particle can communicate only within a small area of the entire search space, such mechanism is local communication. Global communication is implemented when a swarm particle communicates with any particle in the search space, irrespective of its location. In the algorithm, search space in which the weights can vary is limited to [0 1], which is quite small. Local and global communication mechanisms can be considered the same in such a constrained space.Locality can also be defined as the range within which the swarm particle can adjust its weights or thresholds. ACO model's pheromone sensing range is used as an inspiration in defining such range. Fig 3 shows the ant pheromone sensing range of an ant in a two dimensional space.

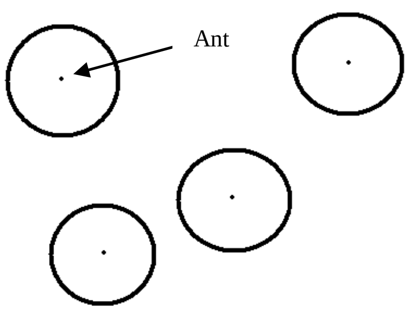


Fig 3: Ant pheromone sensing Range

The ant's pheromone sensing capability is shown in the form of a circle indicating that the ant can sense the chemical (pheromone) presence anywhere within the circle. The circle represents the two-dimensional area within which the ant can change its position [3]. However, the weight adjustments for the swarm particle is one-dimensional i.e. the weight can be increased or it can be decreased. Thus, the particle can be defined to explore the search space in one dimensional model. Fig.4 shows the transformation of ACO sensing range to a one dimensional swarm particle sensing range.

Collision avoidance: Since the swarm particles are adjusting their weights and thresholds independently without interfering in the weight adjustments of its neighbors, collision avoidance is ensured.

Velocity Matching is a key attribute in the swarm approach because it influences the convergence time. Each swarm particle is assigned a velocity component. The velocity component controls the magnitude and direction of the weight change of the swarm particle. The velocity is made directly proportional to the difference in the fitness levels of the swarm particle and the global best or leader. Thus, velocity represents the influence of a swarm leader on a particle. However the swarm particles cannot be influenced on a large scale by its neighbors or in this case, the global leader over its sensing capabilities. Fig 5(a) shows an instance of swarm interaction. The particle 'b' is supposedly at the local maximum forming a temporal leader. The particle 'a' is at a point which is very close to the global best. If the particle 'b' has high influence on the direction of 'a', then 'a' tends to move away from the global maxima towards the local maxima. Fig 5(b) shows the result of such influence allowing 'a' to migrate away from the global best. This kind of influence increases the convergence time for the entire swarm group. To avoid such an effect, a parameter called learning rate is inserted along with the velocity of the particle that guides the influence of swarm leader on a swarm particle. The weight adjustment can be shown in the form of an equation

$$Weight\ adjustment = initial\ weight + \alpha * (\beta) * (\delta) \qquad (1)$$

$\alpha = learning\ rate$

β=fitness of the swarm leader-fitness of the swarm particle

δ=weight of the swarm particle-weight of the swarm leader

β denotes the speed term, which defines the speed with which the particle has to travel towards the leader.

δ denotes the direction in which the weight has to be modified.

Thus β*δ defines the velocity of the swarm particle.

4. SWARM ALGORITHM

Fig 6 shows the flow chart of the swarm algorithm. By using the techniques described in the previous sections and the corresponding observations, the swarm algorithm for the Iris data classification can be defined as repetition of seven steps.

Step 1: The swarm particles with identical behavioral model are created.

Step2: The fitness, inverse of the mean square classification error is calculated for all the swarm particles.
Higher the mean error lower is the fitness of the particle higher is the fitness.

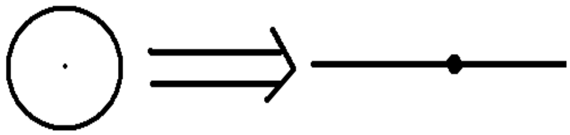


Fig 4: Transformation depicting swarm particles for the one dimensional weight adjustment.

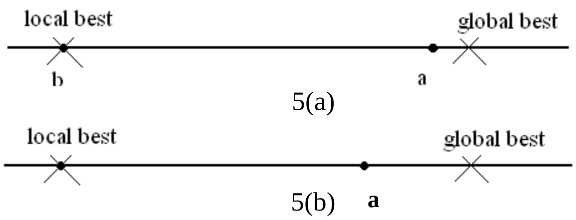


Fig 5: Instance of swarm interaction

Step 3: The swarm particle with highest fitness is identified and declared as the global leader.
The global leader directs the particles towards better classification results
Step 4: The other swarm particles tend to follow the leader each equipped with different velocities. The swarm particle with least fitness is provided with high velocities when compared to the particles with better fitness. This concept is based on the fact that when a particle has higher fitness, there is a higher probability that the optimum solution is lying its close vicinity, so a lower velocity ensures the exploration of its surroundings properly and a particle possessing low fitness has less probability to have optimum solution in its vicinity, so a higher velocity ensures that it converges faster.
Step 6: The leading swarm particle is allowed to explore its surroundings for better fitness.
Step 7: The fitness is calculated for all the particles to decide upon the group leader. It can be observed that the same swarm particle may not be the leader every time.
These seven steps are repeated till the mean fitness reaches acceptable level. However, there is an important procedure that has to be taken care of; adjustment of leader's weights to ensure improving fitness in the consecutive swarm iterations. The principle problem involved in this step is that the swarm particle has no reference to follow like the other particles. Fig 7 shows the weight adjustment for the swarm leader.
Initially the first weight of the swarm leader is considered and a new weight is generated this weight is incremented by a small value 'delta'. The fitness of the swarm leader is calculated with the new weight. If the fitness of the leader is improved, then the new weight is updated and the next weight is considered. If the fitness hasn't improved, then the swarm particle's weight is decreased by delta to generate another weight. The fitness of the swarm leader with this weight is tested. If the fitness is improved, then this weight is updated and the next weight is considered.

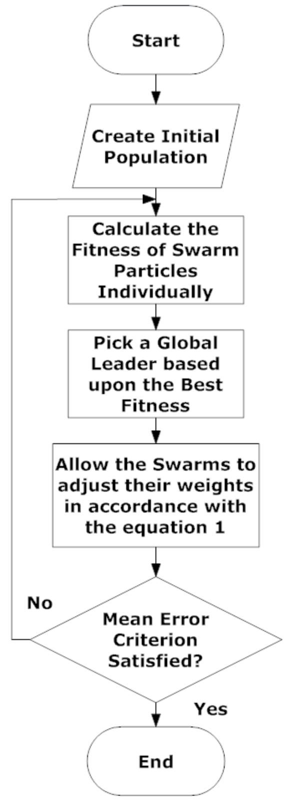


Fig 6: Flow Chart of Swarm Algorithm

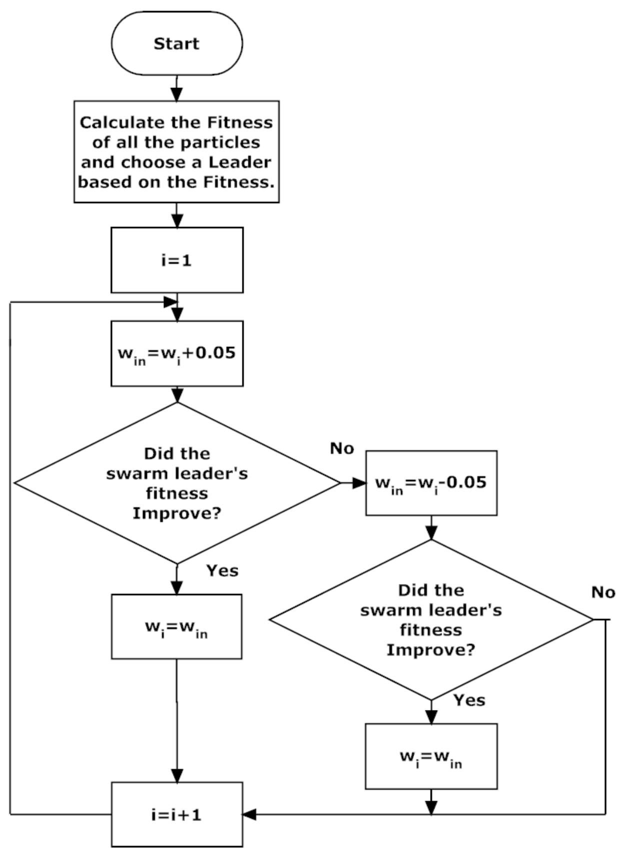
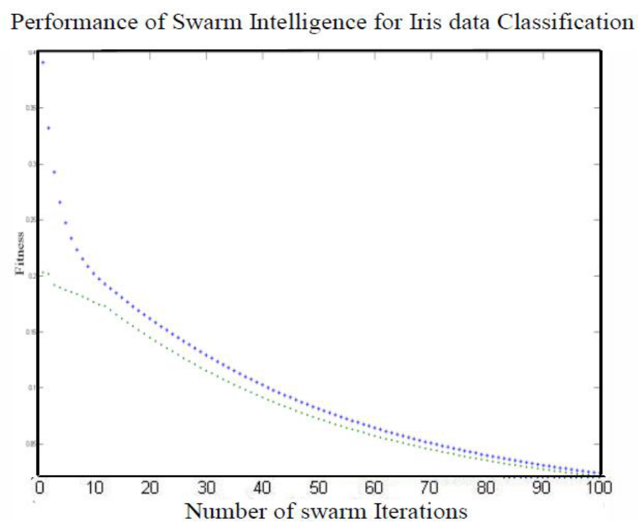
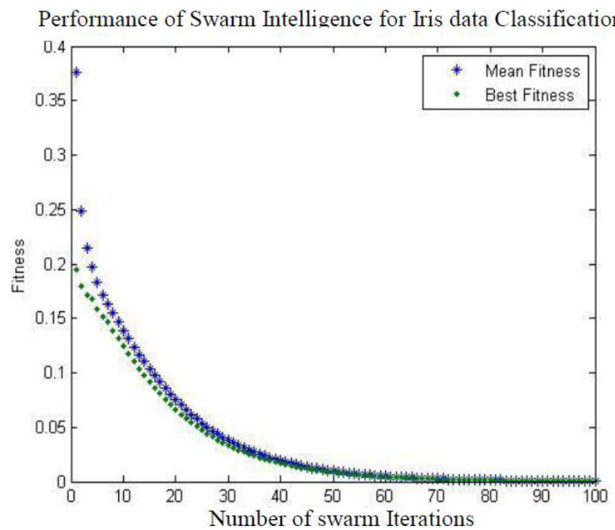


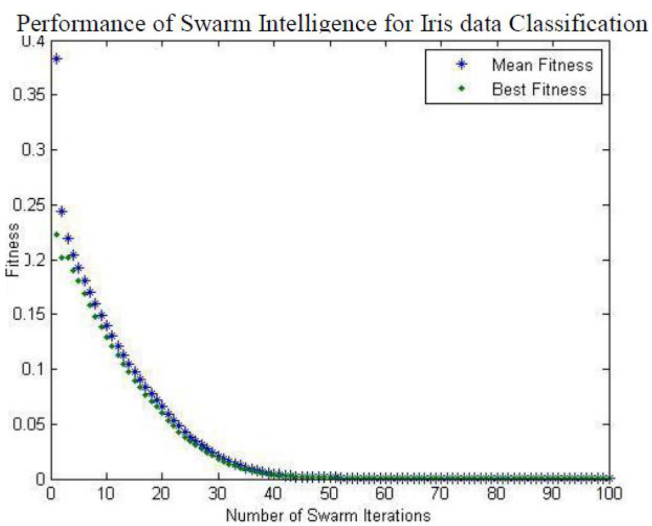
Fig 7: Exploration of surroundings by the swarm leader



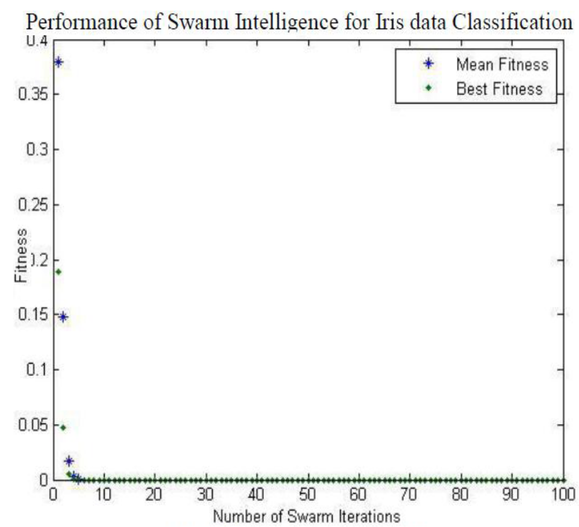
8(c) Learning Rate=0.15



8(c) Learning Rate=0.30



8(c) Learning Rate=0.45



8(d) Learning Rate=0.55

Fig 8: Convergence with Various learning rates.

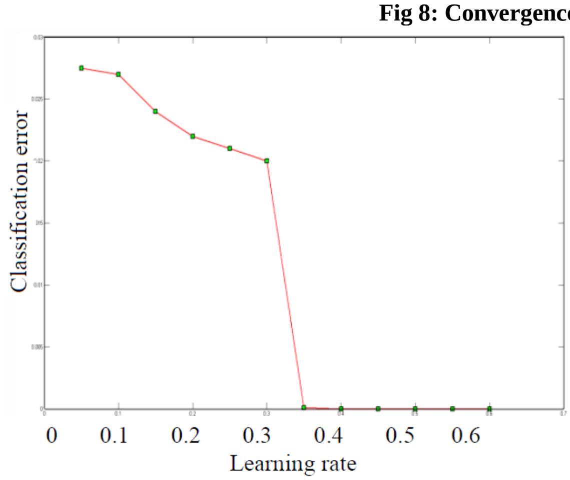


Fig 9: Classification error as function of learning rate

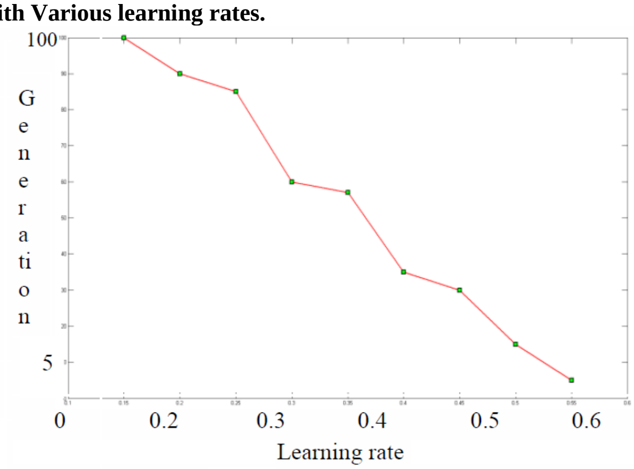


Fig 10: Convergence Graph

Table 2: Comparison of Selectively Cloned NN, Genetic Algorithm and conventional NN Back Propagation Algorithm with swarm algorithm

Technique Used	Number of epochs or generations required for convergence	Best error that could be achieved at the end of convergence	Time Required for the convergence	Converging Time required for the entire program	Time efficiency compared to Back propagation	Convergence efficiency 1 swarm iteration= 1 generation=1 epoch
Swarm Algorithm	**5 iterations**	**0**	**0.1505 sec**	**5.8598 sec**	**97.76%**	**200 %**
Back Propagation Algorithm	**1000 epochs**	**0.02**	**6.739 sec**	**6.739 sec**	**-**	**0.012%**
Genetic Algorithm	**18 generations**	**0(as $1.9x10^{-21}$ is appx as 0)**	**1.45 sec**	**12 sec**	**78.4%**	**55.55%**
Genetic Algorithm with selective cloning	**12 generations**	**0**	**0.272 sec**	**2.086 sec**	**95.96%**	**83.33%**

If the fitness hasn't improved, then the swarm particle's weight is decreased by delta to generate another weight. The fitness of the swarm leader with this weight is tested. If the fitness is improved, then this weight is updated and the next weight is considered, else the weight in consideration is not modified and the next weight is considered to repeat the procedure. In this approach, if a fitness improvement is found in the incremented weight, then the process of checking fitness with the decremented weight is skipped. For instance, a weight when increased by delta may show a fitness improvement of 5%, and the same weight when decreased by delta may show a fitness improvement of 7%. This may lead to performance deterioration. The reason for choosing this kind of approach is that the probability of finding better weights in both the directions (incremental and decremented directions) is very low and also this approach minimizes the convergence time, by not checking for both of the weight operations all the time.

5. RESULTS

In this paper, swarm based algorithm for neural network training has been developed. It has been shown that the learning time of this algorithm is very small in comparison to other neural network training algorithms. As shown in Table 2, the proposed algorithm is 97.76% faster as compared to the back propagation algorithm. Our algorithm outperforms evolutionary neural network learning algorithms. As shown in the Table 2, simple genetic based algorithm for training is only 78.4% faster and genetic algorithm for selective cloning is 95.96% faster in comparison to back propagation algorithm. The swarm based algorithm also reduced the classification error to almost zero. It has been shown that convergence time reduces with the appropriate selection of the learning rate. Fig. 8 shows the convergence rate for various learning rates. It time is not a constraint, it is shown in Fig. 9a that algorithm converges after 50 generations for any learning rate greater than 0.4. Maximum learning happens when learning constant is in the interval [0.3 0.4]. Fig 10 shows the convergence graph, which indicates the converging generations of various learning rates. The converging swarm iterations have been reduced from 90 to 5, when the learning rate is varied from 0.2 to 0.55.

6. REFERENCES

[1] Erdogmus, D. ,Fontenla-Romero, O. Principe, J.C. , Alonso-Betanzos, A.,Castillo, E.,Jenssen, R. Accurate initialization of neural network weights by backpropagation of the desired response. *Proceedings of the International Joint Conference on Neural Networks, IEEE* (July 2003), vol. 3, 2005-2010.

[2] Wei Gao. Evolutionary Neural Network Based on New Ant Colony Algorithm. *International Symposium on Computational Intelligence and Design, IEEE* (Oct 2008), vol. 1, 318 – 321.

[3] Dorigo, M. ,Birattari, M. ,Stutzle, T. Ant colony optimization. *Computational Intelligence Magazine, IEEE* (Nov 2006), vol. 1, 28-39.

[4] Zheping Yan, Yanchao Zhang, Xiaomin Fu ,Shuping Peng. Research of a genetic algorithm ant colony optimization based on cloud model. *International Conference on Mechatronics and Automation, IEEE (*Aug 2009), 4725 – 4730.

[5] Lin Lu, Qi Luo, Jun-yong Liu, Chuan Long. An improved particle swarm optimization algorithm. *International Conference on Granular Computing, IEEE* (Aug 2008), 486-490.

[6] Bing Gong. A Novel Learning Algorithm of Back-Propagation Neural Network. *International Conference on, IEEE* (July 2009), 411-414.

[7] Dorigo, Marco. Swarm-Bots and Swarmanoid: Two Experiments in Embodied Swarm Intelligence. *International Joint Conferences on Web Intelligence and Intelligent Agent Technologies,IEEE* (Sept 2009), 2-3.

[8] S. Haykin. *Neural networks - a comprehensive foundation*. Printice Hall, USA, 2 edition, 1999.